Fast Parallel Iterative Solution of Poisson's and the Biharmonic Equations on Irregular Regions

A. Mayo
A. Greenbaum

Technical Report 540

January 1991

NYU COMPSCI TR-540
Mayo, Anita
Fast parallel iterative
solution of Poisson's and
the biharmonic equations

CO[illegible] UNIVERSITY
ITUTE LIBRARY
New York, N.Y. 10012

Fast Parallel Iterative Solution of Poisson's and the Biharmonic Equations on Irregular Regions

A. Mayo
A. Greenbaum

Technical Report 540

January 1991

Fast Parallel Iterative Solution of Poisson's and the Biharmonic Equations on Irregular Regions

A. Mayo *
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598

A. Greenbaum †
Courant Institute of Mathematical Sciences
251 Mercer Street
New York, NY 10012

January 8, 1991

Abstract

In [10] a method was introduced for solving Poisson's or the biharmonic equation on an irregular region by making use of an integral equation formulation. Because fast solvers were used to extend the solution to an enclosing rectangle, this method avoided many of the standard problems associated with integral equations. The equations that arose were Fredholm integral equations of the second kind with bounded kernels. In this paper we use iterative methods to solve the dense nonsymmetric linear systems arising from the integral equations. Because the matrices are very well-conditioned, conjugate gradient-like methods can be used and will converge very rapidly. The methods are very amenable to vectorization and parallelization, and we describe parallel and vector implementations on shared memory multiprocessors. Numerical experiments are described and results presented for a 3-dimensional interface problem for the Laplacian on a recording head geometry.

*Part of this work was performed while visiting the Courant Institute of Mathematical Sciences.

†This work was supported by the Applied Mathematical Sciences Program of the U.S. Department of Energy under contract DE-AC02-76ER03077.

1 Introduction

In this paper we present efficient parallel numerical methods for solving Poisson's and the biharmonic equations on general regions. Aside from being parallel, these methods also vectorize very well. This is in contrast to other methods for solving these equations on general irregular regions, notably finite element methods. Finite element methods may require use of a fast scatter/gather operation in order to efficiently perform a matrix-vector multiply for the large sparse matrix that arises. In addition, many of the standard preconditioners used in the iterative solution of such equations require inherently sequential operations (e.g., backsolving with the incomplete Cholesky decomposition [12]). This is especially true in the case of the biharmonic equation, and we know of no other effective parallel and vectorizable methods for solving it.

The methods presented here combine integral equation formulations of the problems with rapid finite difference methods on a larger rectangular region in which the irregular region is embedded. Both the integral equation method and the finite difference method parallelize and vectorize well. By using well-conditioned integral equation formulations and iterative methods for solving them, the main part of the computation is reduced to the iterative solution of a dense nonsymmetric matrix equation, instead of a (much larger) sparse (although symmetric) system of equations with an irregular pattern of nonzero elements.

The method used is very similar to the method developed in [10]. The main idea is to use the integral equation formulation to define a discontinuous extension of the solution to the remainder of the embedding region. All the discontinuities between the original function and its extension can be expressed in terms of the solution of the integral equation. These discontinuities are used to compute an approximation to the discrete Laplacian of the combined function at mesh points near the original boundary. Fast Poisson solvers are then used to compute the extended function. This method can also be used to solve directly for the derivatives of the solution, which are usually the physically meaningful quantities.

This combination of integral equation formulations with finite difference methods on a larger rectangular region overcomes two of the standard problems usually associated with integral equation methods. First, because the solution is actually computed using a fast Poisson solver, it is not necessary to compute a costly integral in order to evaluate the solution at many points. Second, although the kernel of the integral is singular, and hence difficult to

evaluate accurately quadrature near the boundary, the solution can be computed very accurately using the fast Poisson solver. In addition, since the methods make use of integral equation formulations, they are very well-suited to problems on exterior regions. A future aim is to use these methods for solving nonlinear magnetostatic problems in unbounded domains.

Work described in the current paper differs from that in [10] in several important respects. One difference is that we use a different integral equation formulation of the biharmonic equation, which has probably not been applied before in numerical computations. The other primary difference is the way in which the integral equations are solved. Conjugate gradient type iterative methods (i.e., the biconjugate gradient /conjugate gradient squared method [5,18]; GMRES [16]; and the conjugate gradient method applied to the normal equations [6]), with simple diagonal preconditioners, are used to solve both Laplace's and the biharmonic equation. Since most of the operations in these methods consist of matrix-vector products, and since the matrices are dense, it is easy to see how the calculations can be performed in parallel and each of the parallel sections can be vectorized. The matrix can be divided into blocks, and each processor can handle multiplication by that block, while vectorization can be performed over the rows within the block. This can actually be accomplished using a level-2 Blas routine [4], and these routines have already been optimized on many parallel/vector supercomputers.

We also note that the integral equations used for solving both of these problems are well-conditioned Fredholm integral equations of the second kind, and so the matrices have eigenvalues and singular values which are bounded independent of the number of discretization points. Therefore we were able to achieve convergence in a small number of iterations. In particular, we were able to solve the biharmonic equation on general nonconvex regions, and, except on very eccentric ellipses, all of the iterative methods tested converged in fewer than 30 iterations. To solve the biharmonic equation on an ellipse with eccentricity 10 or 20, required no more than 32 GMRES iterations, provided enough direction vectors were saved.

In addition to using iterative methods to solve the integral equations, we also solved equations with different types of boundary conditions than before. Specifically, we solved a 3-D exterior interface problem in magnetostatics that arises from modeling a recording head, and a 2-D Dirichlet problem for Laplace's equation on a doubly connected region.

We note that fast Poisson solvers have also been used for solving Poisson's equation on irregular regions through the use of capacitance matrix

methods [2,14]. Our method is, however, better suited to problems on exterior regions. Although, by using a trick due to Hockney [7] and improved by James [8], it is possible to use capacitance matrix methods to solve exterior problems, both the operation count and the storage requirements for the fast solver are significantly increased. More important, capacitance matrix methods do not allow one to solve for derivatives directly. Moreover, we know of no implementation of a capacitance matrix method for solving the biharmonic equation on an irregular region or for solving interface problems.

The organization of the paper is as follows. Section 2 presents the integral equation formulations, and Section 3 the method used for obtaining solutions to the differential equations, given the solutions of the integral equations. Section 4 describes results of numerical experiments carried out on the 8 processor New York University ultracomputer prototype, on the Cray X-MP, and on the IBM 3090.

2 Integral Equations

2.1 Laplace's Equation

Laplace's equation in two dimensions with Dirichlet boundary data $g(t)$ prescribed on the boundary curve $(x(s), y(s))$ is solved in terms of a double layer density function. In this formulation, the solution $u(z)$ is expressed as the integral over the boundary of the region of the product of an unknown density function μ with the normal derivative of the Green's function in the plane:

$$u(z) = \frac{1}{2\pi} \int_{\partial D} \frac{\partial \log r(s,z)}{\partial n_s} \mu(s) ds, \quad z \epsilon D, \tag{1}$$
$$\textit{where} \quad r(s,z) = |s - z|.$$

If the region is simply connected, the density function μ satisfies

$$\mu(t) + \frac{1}{\pi} \int_{\partial D} \frac{\partial \log r(s,t)}{\partial n_s} \mu(s) ds = 2g(t).$$

If the region is doubly connected, with boundary curves L_0 and L_1, and $u(z)$ is the real part of a single valued analytic function, then the density function satisfies the following equation [13]:

$$\mu(t) + \frac{1}{\pi} \int_{\partial D} \left(\frac{\partial \log r(s,t)}{\partial n_s} + a(s,t) \right) \mu(s) ds = 2g(t), \tag{2}$$

$$where \quad a(s,t) \;=\; \begin{cases} 1 & \text{if } s \text{ and } t \text{ both lie on } L_1 \\ 0 & \text{otherwise} \end{cases}$$

These are Fredholm integral equations of the second kind and, if the boundary ∂D is smooth enough, they can be solved very accurately numerically.

Outside D, formula (1) defines another harmonic function $\tilde{u}(z)$:

$$\tilde{u}(z) \;=\; \frac{1}{2\pi}\int_{\partial D} \frac{\partial \log r(s,z)}{\partial n_s}\mu(s)ds. \tag{3}$$

The function $\tilde{u}$ is, of course, a discontinuous extension of the function u. However, all the discontinuities between u and $\tilde{u}$ can be expressed in terms of the density μ and the boundary curve. In Section 3 we show how to determine the discontinuities and how to use them to rapidly compute an approximation to the combined function $U(z)$, which is equal to $u(z)$ inside D, and $\tilde{u}(z)$ outside D.

We can also use formula (1) to express the conjugate function $v(z)$ to $u(z)$ in terms of the density μ. Since the conjugate harmonic function of $\frac{\partial \log r(s,z)}{\partial n_s}$ is $\frac{\partial \log r(s,z)}{\partial s}$, we have

$$v(z) \;=\; \frac{1}{2\pi}\int_{\partial D} \frac{\partial \log r(s,z)}{\partial s}\mu(s)ds. \tag{4}$$

2.2 Interface Problems

A similar formulation can be used to solve certain interface problems. In linear magnetostatics, for example, the following problem arises. One seeks to find a function $u(z)$ defined on a magnetizable region D and on the infinite exterior region outside D such that

$$\nabla \cdot a \nabla u(z) \;=\; \phi(z),$$

where $\phi(z)$ is the given potential function for the applied field, a is the magnetic permeability of the region, and $u(z)$ is the unknown potential function for the H field (i.e., the function satisfying $\nabla u = -H$). In typical linear problems, a is equal to a large constant a_1 inside the magnetizable region D, and equal to a small constant a_0 in the exterior region. Continuity of the H field and the normal component of the B field imply that $u(z)$ and $au_n(z)$ are continuous across the boundary of D.

It turns out that both inside and outside D, the potential u can be expressed as the sum of a term involving the applied potential and the

integral of a double layer density function, where the density is the value of the potential on the boundary:

$$u(z) = \frac{\tilde{a}+1}{\tilde{a}} \int_{\partial D} \frac{\partial G(s,z)}{\partial n} u(s)ds + \frac{\phi(z)}{\tilde{a}}, \quad z \epsilon D \tag{5}$$

$$u(z) = (\tilde{a}+1) \int_{\partial D} \frac{\partial G(s,z)}{\partial n} u(s)ds + \phi(z), \quad z \epsilon D^c \tag{6}$$

where $\tilde{a} = \frac{a_1}{a_0}$ is the relative premeability. See [9]. On the boundary of the region, the potential $u(t)$ satisfies

$$u(t) - \frac{\tilde{a}-1}{\tilde{a}+1} \int_{\partial D} \frac{\partial G(s,t)}{\partial n} u(s)ds = 2\phi(t). \tag{7}$$

In two dimensions, the kernel (Green's function) is $\frac{1}{2\pi} \log r(s,z)$, while in three dimensions it is $\frac{1}{4\pi r(s,z)}$. We have solved this integral equation in three dimensions, in which case the kernel, while in $L^2(\partial D)$, is unbounded.

2.3 The Biharmonic Equation

The integral equation formulation used for the biharmonic equation relies on the representation of a biharmonic function in terms of Goursat functions. Any two dimensional biharmonic function $W(z)$ can be expressed as

$$W(z) = Re(\bar{z}\phi(z) + \chi(z)), \tag{8}$$

where $\phi(z)$ and $\chi(z)$ are analytic functions. The functions $\phi(z)$ and $\psi(z) = \chi'(z)$ are known as the Goursat functions. For a given biharmonic function $W(z)$, the Goursat functions are not uniquely determined. More precisely, $\phi'(z)$ is determined to within a purely imaginary additive constant, and so $\phi(z)$ is determined to within an additive term of the form

$$i\alpha z + \beta,$$

where α is real and β is complex. Similarly, $\psi(z)$ is not uniquely determined. However, all the *physically meaningful quantities*, such as velocities in fluid mechanics or stresses and displacements in elasticity can be expressed in terms of certain derivatives of the Goursat functions, which *are* uniquely determined. Therefore, it suffices to compute these two functions, and not the biharmonic function itself. Both of these functions can be expressed as Cauchy integrals:

$$\phi(z) = \frac{1}{2\pi i}\int_{\partial D}\frac{\omega(s)}{s-z}ds\,, \qquad \psi(z) = \frac{1}{2\pi i}\int_{\partial D}\frac{\bar{\omega}(s)-\bar{s}\omega'(s)}{s-z}ds\,. \quad (9)$$

The integral equation one uses to determine the density function $\omega(t)$ depends on the boundary conditions. We have considered the case in which W_x and W_y (or, equivalently, W and W_n) are prescribed. In previous work [10], the following integral equation, originally developed by Lauricella and Sherman [17], was used:

$$\omega(t) + \frac{1}{\pi}\int_{\partial D}\omega(s)d\theta - \frac{1}{\pi}\int_{\partial D}\bar{\omega}(s)e^{2i\theta}d\theta + \frac{b}{\pi i}Re(\int_{\partial D}\frac{\omega(s)}{(s-c)^2}ds) = f(t), \quad (10)$$

where

$$f(t) = W_x(t) + iW_y(t),$$

$$\theta(s,t) = \arctan(\frac{y(s)-y(t)}{x(s)-x(t)}), \quad b = \frac{1}{t-c} - \frac{1}{(\bar{t}-\bar{c})^2} + \frac{t-c}{(\bar{t}-\bar{c})^2},$$

and c is any point inside the region D. It is important to note that the kernel of this equation is always bounded.

It is known [13] that if a function $\omega(t)$ satisfies this equation, then the third integral in the equation is zero, provided $f(t)$ satisfies the natural compatibility condition $Re(\int_{\partial D}\bar{f}(t)dt) = 0$. Thus equation (10) can be replaced by

$$\omega(t) + \frac{1}{\pi}\int_{\partial D}\omega(s)d\theta - \frac{1}{\pi}\int_{\partial D}\bar{\omega}(s)e^{2i\theta}d\theta = f(t). \quad (11)$$

We chose to solve this integral equation instead of (10), despite the fact that this equation need not have a unique solution. This is acceptable for the following reason. Any solution $\omega_0(t)$ of the homogeneous equation

$$\omega_0(t) + \frac{1}{\pi}\int_{\partial D}\omega_0(s)d\theta - \frac{1}{\pi}\int_{\partial D}\bar{\omega}_0(s)e^{2i\theta}d\theta = 0$$

corresponds to the Goursat functions $\phi_0(z) = i\alpha z$ and $\psi_0(z) = 0$, where α is real [13]. Since $\phi(z)$ is only determined up to terms of this form, and since the physically meaningful quantities are not changed by adding terms of this type to $\phi(z)$, *any* solution of the modified equation will provide a physically correct solution. Existence of multiple solutions is not a problem

for the iterative methods we used, as will be explained in Section 4. Furthermore, it is easy to show that whenever the kernel $\frac{\partial \theta}{\partial s}$ is symmetric, the kernel of (11) will be symmetric. This is true, for example, when the region is an ellipse. A symmetric kernel is an advantage, since in this case the minimum residual variant of the conjugate gradient algorithm can be used to solve the problem (and, in fact, the biconjugate gradient method and the (unrestarted) GMRES method reduce to the minimum residual conjugate gradient algorithm). Conjugacy of direction vectors is maintained without having to save vectors and explicitly orthogonalize. In addition, even when the kernel is not symmetric, we found that the kernel of (11) is closer to being normal than that of (10). Experimental evidence indicated that convergence was always more rapid with equation (11) than with equation (10), so it was decided to use this modified equation.

3 Solving the Differential Equations

After solving the integral equations of Section 2 for the density function μ, we are then able to compute the solution $u(z)$, at points z within the region D, in the following manner. Discontinuous extensions of the solutions to the differential equations are defined throughout a larger embedding rectangular region. Specifically, equation (1) defines the extension of the solution of Laplace's equation, and equation (9) defines the extensions of the real and imaginary parts of the Goursat functions for the biharmonic equation. These functions are all harmonic, except at the boundary of the original region. Therefore, their discrete Laplacians are all zero, up to truncation error, except at mesh points near the boundary. We compute approximations to the discrete Laplacians at these points near the boundary and approximations to the extended function at the edge of the embedding region. Then we apply fast Poisson solvers to obtain approximate solutions to the differential equations, throughout the domain D.

3.1 Solution of Laplace's Equation

The integral equation (1) can be used with the second-order accurate five-point discrete operator Δ_h^5 to compute a second order accurate solution to $\Delta u = 0$ on an irregular region D with smooth boundary $\partial D = (x(s), y(s))$ on which smooth Dirichlet boundary data $u(s) = g(s)$ is prescribed. The procedure is as follows.

First embed D in some regular region R, such as a square with a uniform mesh of width h in the x and y directions. Define a discontinuous extension $\tilde{u}$ of u throughout the region R, using formula (3). As noted previously, all the jump discontinuities in the derivatives of u and $\tilde{u}$ can be expressed in terms of the density function and the derivatives of the boundary curve. Once the discontinuities in the derivatives of the combined function are known and the distances from the boundary curve to the neighboring mesh points are computed, it is easy to compute the discrete Laplacian of the combined function. Knowing an approximation to the discrete Laplacian throughout the regular region R, a fast Poisson solver can then be used to obtain an approximation to the solution u.

Define $U(z)$ to be the combined function,

$$U(z) = \begin{cases} u(z) & z \in D \\ \tilde{u}(z) & z \in D^c \end{cases}$$

The discontinuity between u and $\tilde{u}$ at a point on the boundary of D is equal to the value of the density at that point [13]. Therefore, the discontinuity between their tangential derivatives is equal to the derivative of the density,

$$u_s - \tilde{u}_s = \frac{d\mu}{ds}.$$

There is no discontinuity between their normal derivatives:

$$u_n = \tilde{u}_n.$$

Using these two facts and the direction of the curve, we can compute the discontinuities between u_x and $\tilde{u}_x$ and between u_y and $\tilde{u}_y$:

$$u_x(s) - \tilde{u}_x(s) = \frac{\mu'(s)x'(s)}{x'(s)^2 + y'(s)^2}, \quad u_y(s) - \tilde{u}_y(s) = \frac{\mu'(s)y'(s)}{x'(s)^2 + y'(s)^2}. \quad (12)$$

Discontinuities in higher order derivatives of U can be obtained by differentiating these expressions.

In two dimensions it is also possible to compute these discontinuities by using the fact that $U(z)$ is the real part of a Cauchy integral with density function μ. Let

$$f(z) = \frac{1}{2\pi i} \int_{\partial D} \frac{\mu(\zeta)}{\zeta - z} d\zeta.$$

The kernel in (1) is the real part of the Cauchy kernel:

$$Re(\frac{1}{2\pi i}\frac{d\zeta/ds}{\zeta - z}ds) = \frac{1}{2\pi}\frac{y'(s)[x(s)-x(t)] - x'(s)[y(s)-y(t)]}{(x(s)-x(t))^2 + (y(s)-y(t))^2}ds \quad (13)$$
$$= \frac{1}{2\pi}\frac{\partial \log r(s,z)}{\partial n_s}ds,$$
$$where \quad \zeta(s) = x(s) + iy(s), \qquad z(t) = x(t) + iy(t).$$

Therefore $u(z) = Re(f(z))$ for z in D, and $\tilde{u}(z) = Re(f(z))$ for z outside D. To find the discontinuities between u and $\tilde{u}$, we use the known discontinuities of Cauchy integrals across the boundary curve and the fact that Cauchy integrals are analytic functions. For details see [10].

These discontinuities are then used to compute approximations to the discrete Laplacian at mesh points near the boundary of D. Define the mesh function $U_{i,j}$ by

$$U_{i,j} = \begin{cases} u(x_i, y_j) & (x_i, y_j)\epsilon D \\ \tilde{u}(x_i, y_j) & (x_i, y_j)\epsilon D^c \end{cases}$$

An approximation to $\Delta_h^5 U_{i,j}$ at all mesh points of R can be computed as follows. At mesh points (i,j) of R which have all four of their neighboring mesh points $(i+1,j)$, $(i-1,j)$, $(i,j+1)$, and $(i,j-1)$ on the same side of ∂D, set $\Delta_h^5 U_{i,j} = 0$, since u and $\tilde{u}$ are harmonic. Consider the set B^+ of mesh points which have neighboring mesh points on both sides of ∂D. This set consists of two rings of mesh points, one inside and one outside D. Let p, for example, be a mesh point which is in D but whose neighbor to the right, p_E, is not. Let p^* be the point on ∂D on the line between p and p_E, let h_1 be the distance between p and p^*, and let $h_2 = h - h_1$.

By manipulating the Taylor series at p and p_E, both evaluated at p^*, one can derive the following expression for $\tilde{u}(p_E) - u(p)$. (For details see [10].)

$$\begin{aligned} \tilde{u}(p_E) - u(p) = {} & [\tilde{u}(p^*) - u(p^*)] + h_2[\tilde{u}_x(p^*) - u_x(p^*)] + \\ & \frac{1}{2}h_2^2[\tilde{u}_{xx}(p^*) - u_{xx}(p^*)] + \frac{1}{6}h_2^3[\tilde{u}_{xxx}(p^*) - u_{xxx}(p^*)] + \\ & \frac{1}{24}h_2^4[\tilde{u}_{xxxx}(p^*) - u_{xxxx}(p^*)] + \frac{1}{120}h_2^5[\tilde{u}_{xxxxx}(p^*) - u_{xxxxx}(p^*)] + \\ & hu_x(p) + \frac{1}{2}h^2u_{xx}(p) + \frac{1}{6}h^3u_{xxx}(p) + \frac{1}{24}h^4u_{xxxx}(p) + \\ & \frac{1}{120}h^5u_{xxxxx}(p) + O(h^6). \end{aligned} \quad (14)$$

Note that the first six terms depend on the discontinuities between u and $\tilde{u}$ and their derivatives at the boundary. The other terms are the usual Taylor series terms. Therefore, once these discontinuities are known, the right hand side of (14) is just the sum of known quantities and Taylor series terms.

Now let p_W be the point to the left of p. Then we have

$$U(p_W) - U(p) = \{known\ quantities\} - hu_x(p) + \frac{1}{2}h^2u_{xx}(p) - \frac{1}{6}h^3u_{xxx}(p) + \frac{1}{24}h^4u_{xxxx}(p) - \frac{1}{120}h^5u_{xxxxx}(p) + O(h^6),$$

where the quantities in braces are zero if p_W is in D. In any case, we have

$$U(p_W) + U(p_E) - 2U(p) = \{known\ quantities\} + h^2u_{xx}(p) + \frac{1}{12}h^4u_{xxxx}(p) + O(h^6). \quad (15)$$

If p_N is the point above p and p_S is the point below p, then by the same arguments we have

$$U(p_N) + U(p_S) - 2U(p) = \{known\ quantities\} + h^2u_{xx}(p) + \frac{1}{12}h^4u_{xxxx}(p) + O(h^6). \quad (16)$$

Let $g^+_{i,j}$ denote the sum of the quantities in braces in (15) and (16). Adding (15) and (16) and using the fact that $u_{xx}(p) + u_{yy}(p) = 0$, we obtain

$$h^2\Delta^5_h U(p) = g^+_{i,j} + \frac{h^4}{12}[u_{xxxx}(p) + u_{yyyy}(p)] + O(h^6). \quad (17)$$

The same equation holds with u replaced by $\tilde{u}$ if p is a point outside D since $\tilde{u}$ is then harmonic at p. (It is important to notice that it is *not* assumed that either of the harmonic functions u or $\tilde{u}$ can be extended beyond ∂D. If, however, they could both be extended one mesh width, then the formulas we would obtain for the discrete Laplacian clearly agree with those we have derived.)

Therefore, if the integral equation can be solved accurately enough, one can obtain a fourth order accurate approximation to $h^2\Delta^5_h U(p)$ at points of B^+. This guarantees the accuracy of the solution obtained after applying a fast solver. Define U^h_1 to be the solution of the following set of equations

$$
\begin{aligned}
(\Delta_h^5 U_1^h)_{i,j} &= 0 \quad (x_i, y_j) \in R - B^+ \\
(\Delta_h^5 U_1^h)_{i,j} &= g_{i,j}^+ \quad (x_i, y_j) \in B^+ \\
(U_1^h)_{i,j} &= U_{i,j} \quad (x_i, y_j) \in \partial R
\end{aligned} \tag{18}
$$

where the boundary values $U_{i,j}$ on ∂R are computed by evaluating the integral (3) using the computed density μ. In practice, we have used the trapezoid rule as the quadrature formula for 2-D problems.

If the values of $g_{i,j}^+$ and $U_{i,j}$ are sufficiently accurate, then U_1^h will be a second order accurate approximation to U. For a proof see [10]. By using a higher order accurate approximation to the Laplacian and retaining more terms in the Taylor series, a higher order accurate solution can be obtained.

3.1.1 Computation of the Conjugate Function

The conjugate of a harmonic function can also be computed at small additional cost. Using the Cauchy Riemann equations, the discontinuities in the conjugate function v can be expressed in terms of the discontinuities in u. For example,

$$
v_y - \tilde{v}_y = -(u_y - \tilde{u}_y) = \frac{\mu'(s)y'(s)}{x'(s)^2 + y'(s)^2}.
$$

These discontinuities could also be computed using the fact that v is the imaginary part of the same Cauchy integral that u determines:

$$
v(z) = Im\left(\frac{1}{2\pi}\int_{\partial D}\frac{\mu(\zeta)}{\zeta - z}ds\right) = \frac{1}{2\pi}\int_{\partial D}\frac{\partial \log r(s,z)}{\partial s}\mu(s)ds.
$$

Knowing these discontinuities, we can, in the same way as before, compute an approximation to the discrete Laplacian of v, and then apply a fast Poisson solver to obtain an approximation to v.

3.1.2 Computation of Derivatives

An important property of these methods is that one can easily compute the derivative of a harmonic function directly, without having to compute the function itself. This follows because all the derivatives of a harmonic function are themselves harmonic and because one can compute the discontinuities in all these derivatives.

3.2 Solution of Interface Problems

Since the integrals in (6) and (7) are also integrals of double layer density functions, the same method can be used to solve interface problems.

3.3 Solution of the Biharmonic Equation

As noted in Section 2, the solution of the biharmonic equation can be reduced to the evaluation of two analytic functions which satisfy (9) and the prescribed boundary conditions. More precisely, the evaluation of the physically meaningful quantities reduces to the evaluation of the Goursat functions (8).

Both of the Goursat functions can be expressed as Cauchy integrals whose densities are given in terms of the solution of the integral equation (11). Therefore, it suffices to be able to evaluate a Cauchy integral $g(z) = \int_{\partial D} \frac{f(t)}{z-t} dt$ with prescribed density function $f(t)$. Suppose $f(t) = p(x,y) + iq(x,y)$. The integral $g(z)$ can be expressed in terms of certain integrals of double layer density functions and their conjugates:

$$g(z) = \frac{1}{2\pi i} \int_{\partial D} p(s)\left(\frac{\partial \log r}{\partial n_s} + i\frac{\partial \log r}{\partial s}\right)ds + \frac{1}{2\pi} \int_{\partial D} q(s)\left(\frac{\partial \log r}{\partial n_s} - i\frac{\partial \log r}{\partial s}\right)ds.$$

Therefore, the same methods can be used to evaluate the Goursat functions.

4 Numerical Experiments

In this section we report on calculations we have performed. Parallel implementation of the method is discussed, and details are given on the performance of the iterative methods used for solving the integral equations. Accuracy of the method was discussed in detail in [10], and will not be repeated here.

The complete solution method consists of the following steps:

1. The irregular region D is embedded in a rectangle R with a uniform mesh, and distances from neighboring mesh points to the boundary are computed.

2. The boundary(ies) of the region D are discretized, and the integral operators are replaced by quadrature formulae. Since the kernels are bounded for the 2-D problems, we used the trapezoid rule, since it is

extremely accurate on periodic regions. For the 3-D interface problem, a different procedure was used since the kernel is unbounded. We triangulated the surface of the region, which was flat. We assumed a continuous, piecewise linear density function and integrated the product of the kernel function with the piecewise linear basis functions exactly. Explicit quadrature formulae can be found in [9]. (These formulae require the surface to be flat.)

3. Having formed the dense, nonsymmetric matrix arising from the integral equation, the appropriate linear system is then solved for the density μ using an iterative method. Methods that we experimented with include the biconjugate gradient / conjugate gradient squared method [5,18], GMRES [16], and the conjugate gradient method applied to the normal equations [6]. In each case, the diagonal of the matrix was used as a preconditioner.

4. Approximations to the discrete Laplacian are computed near the boundary ∂D, and approximations to the extended function are computed along the boundaries of R.

5. A fast Poisson solver is applied to obtain an approximate solution to the differential equation.

4.1 Test Problems and Performance of Iterative Methods

Although the matrices that arise from the integral equations of Section 2 are dense and nonsymmetric, they are also well-conditioned and, in most cases, have tightly clustered eigenvalues and singular values. For this reason, all of the nonsymmetric conjugate gradient-type iterative methods that we have tried have performed well. In each case, the diagonal of the matrix was used as the preconditioner. Results are presented primarily for the biconjugate gradient (BCG) method, but some of the problems were also solved using GMRES [16], conjugate gradients on the normal equations (CGNR) [6], and the conjugate gradient squared (CGS) method [18].

The effectiveness of these iterative methods for solving the double layer integral equation for Laplace's equation on simply connected regions is well-documented. (See, for example, [11].) We have performed calculations on doubly connected regions and have found the results to be similar. In particular, we have tested the biconjugate gradient method on a region bounded by two nonconcentric ellipses. The outer ellipse had semiaxes .35 and .40,

n	number of BCG Iterations
50	6
100	6
200	6
360	9

Table 1: Doubly Connected Region 1.

n	number of BCG Iterations
50	9
100	9
200	9
360	12

Table 2: Doubly Connected Region 2.

the inner ellipse had semiaxes .08 and .10, and the center of the inner ellipse was offset .14 from the center of the outer ellipse along the major axis. Starting with zero as an initial guess, we iterated until the relative size of the pseudo-residual was reduced below 10^{-6}:

$$\frac{||M^{-1}r^k||}{||M^{-1}f||} < 10^{-6}.$$

Here M is the diagonal of the matrix, r^k is the residual at step k, and f is the right-hand side of the matrix equation. In all cases, the true residual, $f - Ax^k$ was computed, since, for some problems this may differ from the vector r^k generated by the biconjugate gradient algorithm.

Table 1 shows the number of iterations required to satisfy this convergence criterion, for different values of n, the number of boundary discretization points. Table 2 shows results for a slightly different region. Now the outer ellipse has semiaxes .35 and .15.

The biconjugate gradient method was also used to solve the three dimensional integral equation (7) for an interface problem in magnetics. The region D, shown in Figure 1, is shaped like a U shaped recording head, with equal pole tips of width 5 meters, gap 1 meter, height 30 meters, and uniform depth 10 meters. The function $\phi(z)$ was the applied field due to an infinitely long straight wire with 100 amps of current, and the relative

Figure 1: U Shaped Recording Head

permeability was set to 1000. In one case we used 361 boundary elements and in another case we used 1086 elements. In both cases our initial guess for the potential at a point on the boundary was the value of the applied field at that point. The biconjugate gradient method required 28 iterations to achieve convergence for the smaller problem, 31 iterations for the larger problem.

What is perhaps more interesting is that good results were also obtained using iterative methods to solve the integral equation for the biharmonic equation. Table 3 shows the number of iterations required by various iterative methods to satisfy the convergence criterion on several different regions. We always placed 256 points on the boundary of the region, so the matrix was of order 512. The crescents mentioned in the table are nonconvex and are pictured in Figure 2 and Figure 3. They are given parametrically by

$$x(t) = \cos t + .15 \sin^2 t, \quad y(t) = .5 \sin t + d \cos^2 t.$$

In crescent 1, $d = .4$, and in crescent 2, $d = .7$.

Most of the numbers in the table were obtained by taking the right hand

Region	BCG	CGS	GMRES(5)	GMRES(40)	CGNR
1. circle of radius 1	4	3	4	4	4
2. ellipse with semiaxes 1 and 2	9	6	10	9	12
3. ellipse with semiaxes 1 and 10	32 (20)	27 (18)	73 (58)	32 (20)	74 (41)
4. ellipse with semiaxes 1 and 20	58 (22)	19 (10)	44 (23)	32 (14)	145 (21)
5. crescent 1	11	8	12	11	16
6. crescent 2	16	11	23	15	26

Table 3: Biharmonic Equation (11).

side of the equation to correspond to the biharmonic function $W(x,y) = x^2 + y^2 + x$, and using a *random* initial guess. In practice, however, it is usually advantageous to use a smooth initial guess, such as zero or the right hand side, since the true solution will be similarly smooth. To determine whether a reasonably chosen initial guess could significantly reduce the number of iterations, we solved the biharmonic problem for several more complicated right hand sides, using a zero initial guess. The results did not differ significantly from those obtained previously, except on the highly eccentric ellipses. The numbers in parentheses in Table 3 are iteration counts using an initial guess of zero, when the biharmonic function $W(z)$ is given by

$$W(z) = Re(\bar{z}\phi(z) + \chi(z)), \quad where$$

$$\phi(z) = \sum_{k=0}^{5}(-1)^k\frac{z^k}{k+1} + e^z, \quad \chi(z) = \sum_{k=0}^{5}\frac{z^k}{k+1} - 2e^z.$$

We believe that these convergence rates may be more realistic than those obtained with a random initial guess, for most applications.

Note that the most difficult problems for the iterative methods (with either initial guess) were those defined on highly eccentric ellipses. For problems 3 and 4, the GMRES method converged in 32 iterations (for the random initial guess), provided all direction vectors were saved (GMRES(40)), but required significantly more iterations when only 5 direction vectors were saved (GMRES(5)). Similarly, the other iterative methods were slower to converge on these regions.

It should be noted that some of these iterative methods require more work per iteration than others. Since the bulk of the work at each iteration is in applying the dense matrix to a vector, the BCG, CGS, and CGNR

methods require almost twice as much work per iteration as the GMRES method, even when 40 direction vectors are saved and orthogonalized against at each step. (This is because the matrix is dense and so, a standard matrix vector multiply requires n^2 operations, compared to about $40n$ for orthogonalizing. In future work, it is planned to replace this standard matrix vector multiply routine by an $O(n)$ method [3]. Still, the matrix vector multiply will require significantly more than $40n$ operations, so the comparison of operation counts per iteration for the various iterative methods probably will not change.) Thus, to compare the methods according to total work performed, the iteration counts for BCG, CGS, and CGNR should be multiplied by two and compared with that of GMRES(5) and GMRES(40). It can then be seen that the GMRES(40) method is the most efficient, with CGS or GMRES(5) second. In addition, it may be necessary to compute the residual directly (as we have done) at each step of the CGS algorithm, since the vector r^k computed by updating may not resemble the true residual. If this is done, then the number of matrix vector multiplies per CGS iteration is 3 instead of 2, and the GMRES methods look even better in comparison. Of course, GMRES(40) requires almost 40 vectors more in storage than these other methods, and it is difficult to predict just how many vectors will be needed in order to obtain rapid convergence with GMRES. The least efficient iterative method for these problems is CGNR.

To understand why the iteration counts are as described, we computed the eigenvalues and singular values for these problems. The 512 singular values for each region are plotted in decreasing order of magnitude in Figure 4. The eigenvalues were very close to the singular values (almost indistinguishable on graphs such as Figure 4) and had tiny imaginary parts and nonnegative real parts. This tends to be an advantage for the GMRES method over, say, CGNR, because the convergence rate of GMRES is governed by the eigenvalues while that of CGNR is determined by the *squared* singular values. We note that in all cases, the eigenvalues and singular values cluster around 1. For the eccentric ellipses, more eigenvalues and singular values appear near the ends of the spectrum, causing some difficulty for the iterative methods. The ratio of the largest to the second smallest singular value for each of the six regions is 2.0, 5.2, 265, 2031, 8.5, and 22.3, respectively.

The presence of a zero singular value does not adversely affect the convergence rate of the iterative methods. The reason for this is as follows. For each of these iterative methods, the residual r^k satisfies

$$r^k = P_k(B)r^0,$$

where P_k is a k^{th} or $2k^{th}$ (for CGS) degree polynomial with value one at the origin and $B = A$ for the GMRES, BCG, and CGS methods (actually, $M^{-1}A$, where M is the diagonal of A), $B = AA^T$ for CG on the normal equations. Assuming B has a complete set of eigenvectors V (which it does for each of these problems), we can write $B = V\Lambda V^{-1}$ and

$$V^{-1}r^k = P_k(\Lambda)V^{-1}r^0. \tag{19}$$

Now suppose some eigenvalue λ_i is zero. Since the equations are consistent, the i^{th} component of V^{-1} times the right hand side f of the equation $Bx = f$ must also be zero; for we have

$$\lambda_i(V^{-1}x)_i = (V^{-1}f)_i.$$

It also follows that the i^{th} component of V^{-1} times the initial residual $r^0 = f - Bx^0$ is zero:

$$(V^{-1}r^0)_i = (V^{-1}f)_i - \lambda_i(V^{-1}x^0)_i.$$

It then follows from (19) that the i^{th} component of $V^{-1}r^k$ is zero for all steps k, and it is easy to check that the coefficients computed by each of these iterative methods are the same as those that would be computed if this zero eigenvalue were not present. Thus, these methods converge to *a* solution of the singular linear system at the same rate as they would converge to *the* solution of a nonsingular linear system having only the nonzero eigenvalues.

Table 4 shows the number of iterations needed to satisfy the same convergence criterion, when we discretized equation (10) instead of (11). Note that the number of iterations required was significantly greater for each of the iterative methods. The GMRES(5) method failed on the crescent problems, stagnating (ceasing to reduce the error) well before it reached the desired level of accuracy.

4.2 Parallel and Vector Implementation of the Method

The major part of the work at each iteration of the biconjugate gradient algorithm is in multiplying the matrix and its transpose by an arbitrary vector. This operation is easily vectorized and/or parallelized, as are the other operations (dot products, saxpy's, etc.) required for an iteration. On a machine with both vector and parallel capabilities, different blocks of the matrix can be assigned to different processors and vectorization can be performed over the rows within each block. This can actually be accomplished

Region	BCG	CGS	GMRES(5)	GMRES(40)	CGNR
1. circle of radius 1	8	7	10	8	11
2. ellipse with semiaxes 1 and 2	12	9	15	12	17
3. ellipse with semiaxes 1 and 10	45	35	76	34	92
4. ellipse with semiaxes 1 and 20	86	41	119	50	188
5. crescent 1	30	30	> 200	24	45
6. crescent 2	50	64	> 200	30	82

Table 4: Biharmonic Equation (10) (Random Initial Guess).

using a level-2 Blas routine [4], and these routines have already been optimized on many parallel/vector supercomputers. Another significant part of the work is in applying the fast Poisson solver after the integral equations have been solved. This is also easily parallelized, with different processors computing FFT's simultaneously, while the individual FFT's may be vectorized.

We implemented the entire algorithm for computing a harmonic function, its conjugate, and its first derivatives on an 8-processor shared memory MIMD machine – the NYU Ultracomputer prototype – and obtained good speedup over the serial code. The region was the first doubly connected region bounded by two ellipses described above. We again placed 200 points on the boundary and used a 33 by 33 rectangular mesh on the rectangle in which the region was embedded.

In this case the biconjugate gradient routine accounted for about 40% of the time required by the serial code. To obtain good speedup of the entire code, it was also necessary to parallelize the other sections of the code – generating the matrix, computing the Laplacian at irregular mesh points, and applying the fast Poisson solver. These sections were also easily parallelized, and the use of a parallel DO loop facility, called DOALL, on the ultracomputer allowed efficient parallelization of the entire code with only minor modifications to the original serial code. DOALL prespawns tasks and assigns loop indices to processors dynamically [1]. Different elements of the matrix were generated in parallel and the Laplacian was computed at irregular mesh points in parallel. The 2-D fast Poisson solver consists of 1-D FFT's which were also executed in parallel.

Figure 5 shows the time and speedup on 1 to 8 processors for the above problem. Using 8 processors, an overall speedup of about a factor of 6 was obtained, with the biconjugate gradient routine obtaining about a factor of

7 speedup over its time on 1 processor. The time for the parallel code run on 1 processor was about 5% slower than the time for the serial code. A factor of 7 speedup on the ultracomputer appears to be near the best attainable, due to bus traffic.

On the Cray X-MP, we measured the difference between the time required for the matrix generation only, when vectorized and unvectorized. For this section of the code, vectorization resulted in a factor of 9 speedup, and its effect on the other sections appears to be similar.

Acknowledgment. The authors wish to thank the referees for many helpful comments.

References

[1] W. Berke, ParFOR – A structured environment for parallel Fortran, Ultracomputer Note 137, Courant Institute, New York, 1988.

[2] B.L. Buzbee, F.W. Dorr, J.A. George, and G.H. Golub, The direct solution of the discrete Poisson equation on irregular regions, *SIAM Jour. Num. Anal.*, vol. 8, 1971, pp. 722-736.

[3] J. Carrier, L. Greengard, and V. Rokhlin, A fast adaptive multipole algorithm for particle simulations, *SIAM Jour. Sci. Stat. Comput.*, vol. 9, 1988, pp. 669-686.

[4] J. Dongarra, J. DuCroz, S. Hammarling, and R. Hanson, An extended set of Fortran basic linear algebra subprograms, *ACM Trans. Math. Soft.*, vol. 14, no. 1, 1988, pp. 1-17.

[5] R. Fletcher, Conjugate gradient methods for indefinite systems, in *Proc. of the Dundee Biennial Conference on Numerical Analysis*,

[6] M.R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Standards*, vol. 49, 1952, pp. 409-436.

[7] R.W. Hockney, The potential calculation and some applications, in *Methods in Computational Physics*, Vol. 9, Academic Press, New York, 1970, pp. 135-211.

[8] R. James, The solution of the Poisson equation for isolated source distributions, *Jour. Comp. Phys.*, vol. 25, no. 71, 1977, pp. 71-93.

[9] D. Lindholm, Notes on boundary integral equations for three dimensional magnetostatics, *IEEE Trans. on Magnetics*, vol. 16, no. 6, 1980, pp. 1409-1417.

[10] A. Mayo, The fast solution of Poisson's and the biharmonic equations on general regions, *SIAM Jour. Num. Anal.*, vol. 21, no. 2, 1984, pp. 285-299.

[11] A. Mayo, Rapid, high order accurate evaluation of volume integrals of potential theory, submitted to *Jour. Comp. Phys.*.

[12] J.A. Meijerink and H.A. van der Vorst, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix, *Math. Comp.*, vol. 31, 1977, pp. 148-162.

[13] S.G. Mikhlin, *Integral Equations and their Applications*, Pergamon Press, New York, 1957.

[14] W. Proskurowski and O. Widlund, On the numerical solution of Helmholtz's equation by the capacitance matrix method, *Math. Comp.*, vol. 30, 1976, pp. 433-468.

[15] V. Rokhlin, Rapid solution of integral equations of classical potential theory, *Jour. Comp. Phys.*, vol. 60, no. 2, 1985, pp. 187-207.

[16] Y. Saad and M.H. Schultz, GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems, *SIAM Jour. Sci. Stat. Comp.*, vol. 7, 1986, pp. 856-869.

[17] I. Sherman, The solution of the plane static problem of the theory of elasticity with given external forces, *Dokl. Akad. Nauk SSSR*, 28, 1, 1940.

[18] P. Sonneveld, CGS, a fast Lanczos-type solver for nonsymmetric linear systems, *SIAM Jour. Sci. Stat. Comp.*, vol. 10, 1989, pp. 36-52.

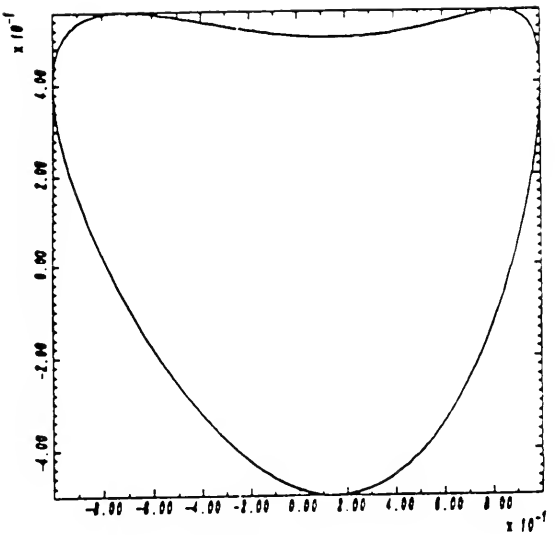


Figure 2

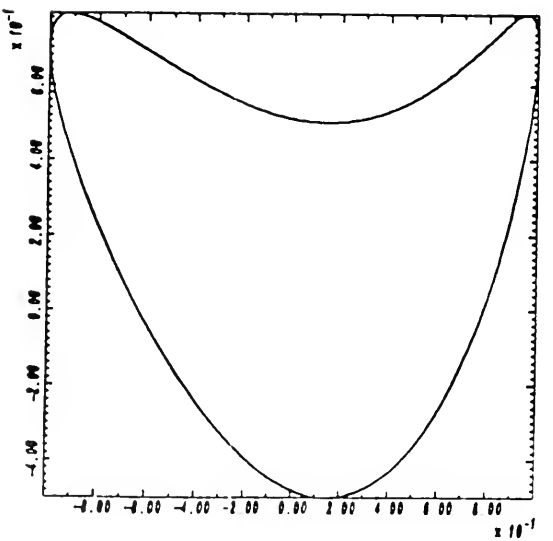


Figure 3

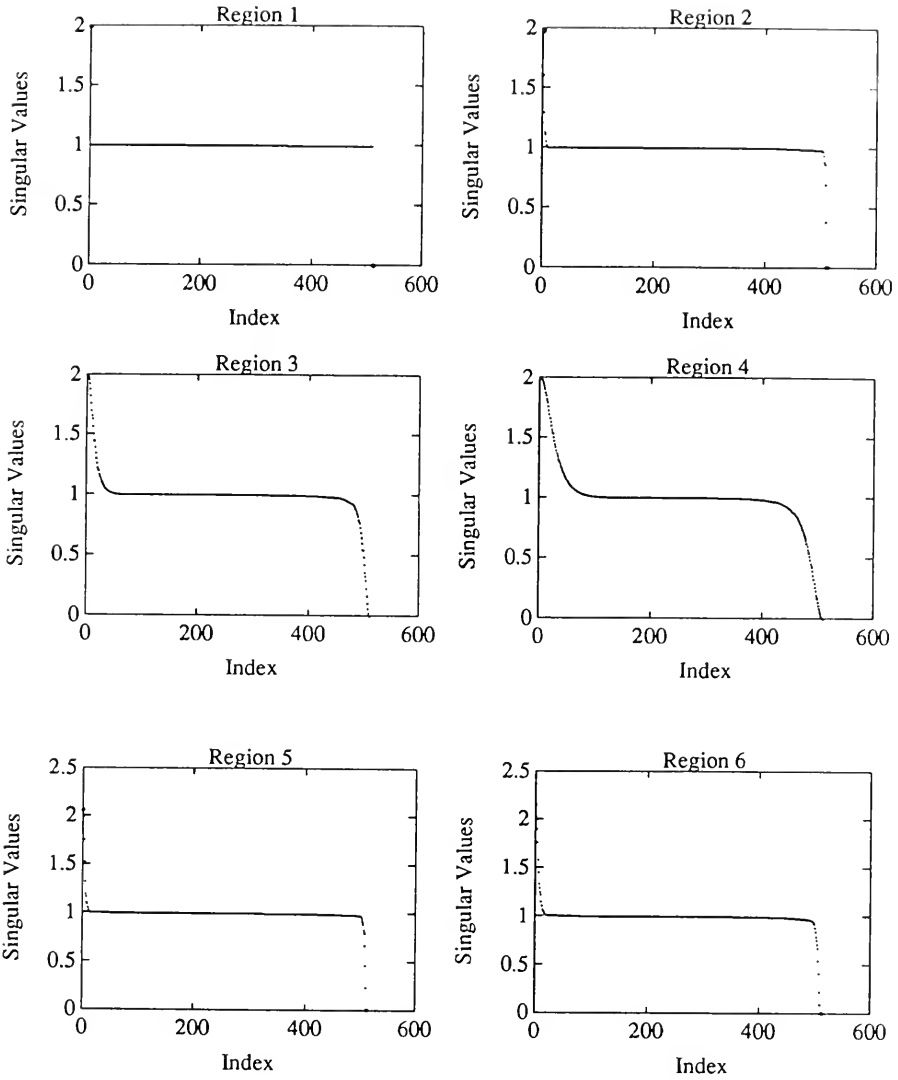
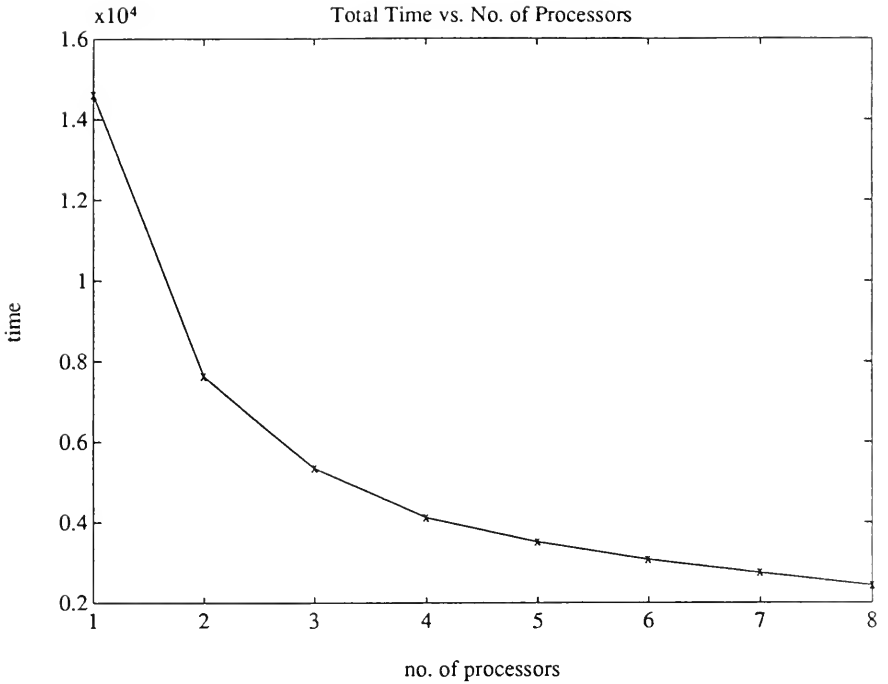

Region 1
Region 2
Region 3
Region 4
Region 5
Region 6
Singular Values
Index
0
200
400
600
0.5
1
1.5
2
2.5

Figure 4


x10⁴
Total Time vs. No. of Processors
time
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
1
2
3
4
5
6
7
8
no. of processors

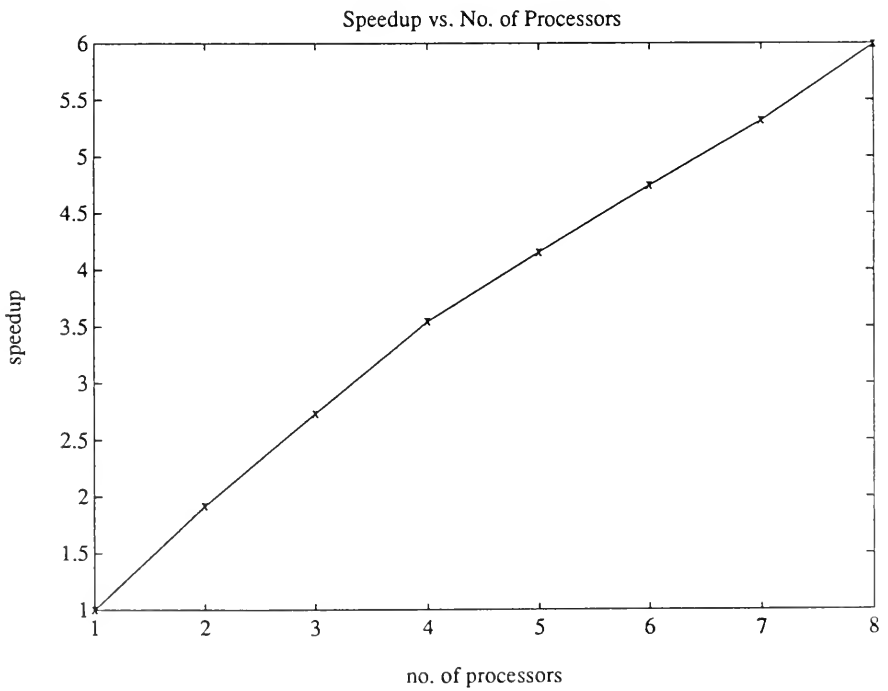

Speedup vs. No. of Processors
speedup
6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
1
2
3
4
5
6
7
8
no. of processors

Figure 5.

NYU COMPSCI TR-540
Mayo, Anita C.1
Fast parallel iterative
solution of Poisson's and
the biharmonic equations

NYU COMPSCI TR-540
Mayo, Anita C.1
Fast parallel iterative
solution of Poisson's and
the biharmonic equations

DATE DUE	BORROWER'S NAME
	June-Yub Lee

DATE DUE

AUG 15 1991

GAYLORD PRINTED IN U.S.A.